iG/Check

for

INTERLIS 2

Inhaltsverzeichnis

1. Einleitung

Das hier vorliegende Benutzerhandbuch beschreibt **iG/Check for INTERLIS 2** für die Betriebsysteme Windows XP, Windows 2003, Windows Vista, Windows 7 und Windows 2008. Bei iG/Check for INTERLIS 2 (im Folgenden auch kurz mit iG/Check bezeichnet) handelt es sich um ein Qualitätssicherungswerkzeug mit dem INTERLIS 2.2 / 2.3 Transferdateien auf Konsistenz gegenüber einem gegebenen INTERLIS 2.2 / 2.3 Datenmodell überprüft werden können. iG/Check kann von Auftraggebern (Bund, Kantone, Gemeinden, Werke) für die Verifikation bzw. von Datenlieferanten (Geometer, Ingenieurbüros) für die systemneutrale Überprüfung ihrer Datenprodukte eingesetzt werden.

Das Benutzerhandbuch ist wie folgt aufgebaut:

- In Kapitel 1 wird auf die Funktionsweise von iG/Check eingegangen.
- In Kapitel 2 ist die Installation von iG/Check beschrieben.
- In Kapitel 3 ist die Bedienung von iG/Check beschrieben.
- In Kapitel 4 ist dokumentiert, welche Fehler von iG/Check gemeldet werden.
- In Kapitel 5 wird auf die Konfigurationsmöglichkeiten eingegangen.

iG/Check for INTERLIS 2 ist Bestandteil des Produktes **INTERLIS Tools**, kann aber auch einzeln lizenziert werden.

Konventionen:

fett	Definitionen, neue Begriffe.
`courier`	Eingaben im Betriebssystem oder Dateien.
kursiv	Datei- und Programmnamen.
[1]	Verweis auf das Literaturverzeichnis im Anhang.

1.1 Funktionsweise

iG/Check benutzt den offiziellen INTERLIS 2 Compiler der KOGIS und kann dadurch die INTERLIS-Modelldatei (*.ili*) und die INTERLIS-Transferdatei (*.xtf*) überprüfen (s.a. 1.2). Da iG/Check die INTERLIS-Modelldatei bei jeder Überprüfung interpretiert, kann *jedes* mit INTERLIS 2 (Version 2.2 oder 2.3) beschriebene Datenmodell, d.h. DM01 Bundesmodell oder auch *jedes* kantonale oder *jedes* benutzerdefinierte Modell von iG/Check überprüft werden. Eine Anpassung von iG/Check an kantonale Mehranforderungen oder Benutzermodelle ist daher nicht notwendig.

1.2 Begriffe

INTERLIS

Eine ursprünglich im Auftrag der eidgenössischen Vermessungsdirektion entwickelte Datenmodellierungssprache (DML) [1]. INTERLIS eignet sich besonders für die Beschreibung von Datenmodellen von geographischen Informationssystemen (GIS). INTERLIS kann aber auch für andere Anwendungen (z.B. Datenbankapplikationen) eingesetzt werden.

DM01

Ein in INTERLIS definiertes Datenmodell für die amtl. Vermessung [2].

AVS

Amtliche Vermessungs Schnittstelle. Jedes in INTERLIS beschriebene Datenmodell definiert automatisch ein Transferformat für den Datenaustausch. Die AVS ist das Transferformat für das Datenmodell DM01.

INTERLIS-Modelldatei

Die INTERLIS-Modelldatei (Endung *.ili*) ist eine ASCII-Datei in der ein konkretes INTERLIS-Datenmodell in INTERLIS beschrieben ist.

INTERLIS-Transferdatei

Eine nach den Regeln des Transferformats aufgebaute XML-Datei (Endung *.xtf*), s.a. [1].

Transferidentifikation

Jedem Objekt in der Transferdatei ist eine eindeutige Transferidentifikation (TID) zugeordnet.

2. Installation

2.1 Überblick

iG/Check wird für alle unterstützten Betriebssysteme gleich installiert. Die folgende Installationsanleitung gilt daher für alle Betriebssysteme.

2.2 Installationsschritte

1. Melden Sie sich unter einem Benutzer mit Administrator Privileg im System an. Die Anmeldung als Administrator ist notwendig, weil das Installationsprogramm Einträge in die Windows-Registratur macht.

2. Installation von *iG/Check for INTERLIS 2*. Legen Sie die mitgelieferte Installations-CD in Laufwerk `d:` und starten Sie das Installationsprogramm mit

   `d:\igcheck2\install`

3. Lesen Sie die Lizenzvereinbarung und stimmen Sie der Lizenzvereinbarung zu, oder brechen Sie die Installation ab.

4. Geben Sie für das Installationsverzeichnis einen gültigen Pfadnamen an (Für iG/Check for INTERLIS 2 empfehlen wir *c:\igcheck2*). Im Weiteren wird dieses Installationsverzeichnis mit `IGCHECK_DIR` bezeichnet. Hinweise: Bitte verwenden Sie nach Möglichkeit keine Installationsverzeichnisse mit Leer- oder anderen Sonderzeichen.

   Das Installationsprogramm kopiert nun die Programmdateien in das Installationsverzeichnis `IGCHECK_DIR` und installiert das Produkt mit dem ICS Runtimesystem (ICS = infoGrips Conversion System) auf ihrem Computer.

2.3 Lizenzierung

Bemerkung: Dieser Schritt muss für den von KOGIS verteilten Checker nicht durchgeführt werden, weil die Lizenz bereits im Installationsprogramm enthalten ist.

1. Rufen Sie das Lizenzierungsprogramm auf:

   `Start>Programme>infoGrips INTERLIS Tools>License`

2. Je nachdem, ob Sie das vollständige Produkt *INTERLIS Tools* oder nur iG/Check lizenziert haben, müssen Sie die Lizenzinformationen in eines der folgenden Lizenzfiles eintragen:

   INTERLIS Tools: `iltools.lic`
   iG/Check: `igcheck2.lic`

   Wählen oder geben Sie mit `Select/Enter` das entsprechende Lizenzierungsfile ein.

3. Tragen Sie die Lizenzinformationen `Company`, `Text`, `Modules` und `Serialno` ein. Die Eingaben müssen **absolut identisch** (d.h. ohne zusätzliche Leerzeichen) zu den Angaben im Begleitbrief sein, da sonst ICS die Lizenzinformation nicht anerkennt.

4. Speichern Sie Ihre Eingaben mit `Save`. Sie können jederzeit die Felder `Company`, `Modules`, `Text` und `Serialno` frisch eingeben, Sie müssen jede Änderung mit `Save` sichern.

5. Beenden Sie mit `Cancel` das Lizenzierungsprogramm.

2.4 Testen der Installation

Nach der Installation von iG/Check 2 können Sie die mitgelieferte INTERLIS-Transferdatei *roads23.xtf* (s.a. [1]) wie folgt prüfen:

1. Starten Sie ICS for Windows über den Windows Start Knopf mit:

   Programme>infoGrips INTERLIS Tools>iGCheck for INTERLIS 2.
2. Starten Sie iG/Check mit **`Run`** und wählen Sie die Datei *\examples\roads23.xtf* aus.
3. iG/Check überprüft nun ob die Transferdatei *roads23.xtf* mit den Spezifikationen in den INTERLIS-Modelldateien unter IGCHECK_DIR\user\models23*.ili bzw. IGCHECK_DIR\system\models23*.ili übereinstimmt.
4. Die Fehlermeldungen können mit **`Show Log`** angezeigt werden.

2.5 Aktualisieren der Installation

Seit Version 2012 kann die iG/Check Installation auch halbautomatisch via den Update-Server der infoGrips GmbH aktualisiert werden. Dazu muss lediglich der Script *\system\script\util\update.cfg* mit ICS for Windows geladen und gestartet werden.

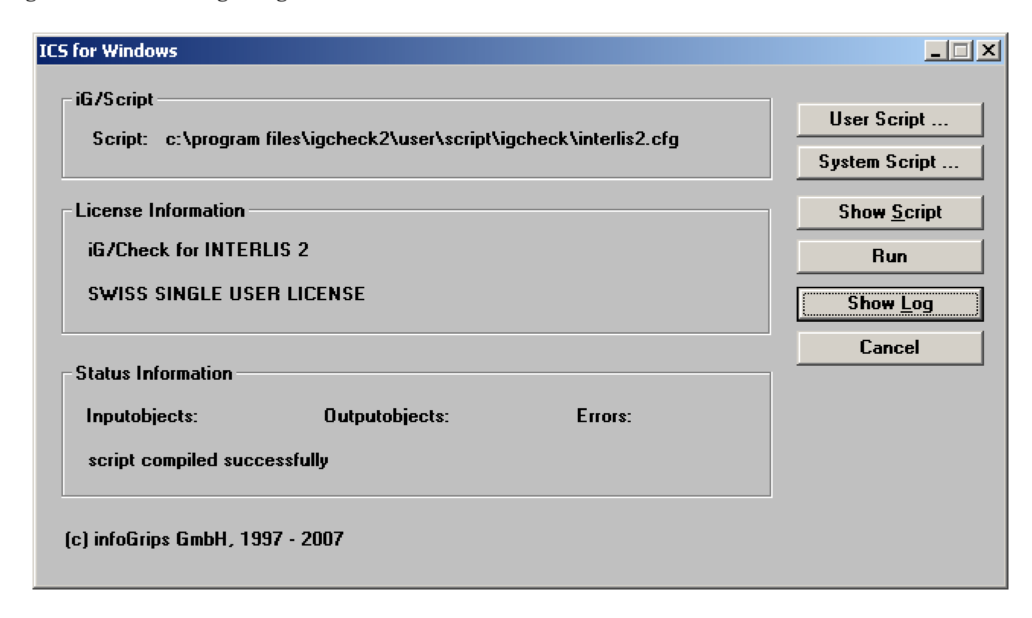
3. Bedienung

iG/Check kann über die Windowsoberfläche *ICS for Windows* oder über die *iG/Check Kommandozeile* aufgerufen werden. Nachfolgend sind beide Umgebungen beschrieben.

Hinweis: iG/Check for INTERLIS 2 unterstützt auch die ältere INTERLIS Version 2.2. Da die Bedienung für Version 2.2 und 2.3 grundsätzlich gleich ist, sind alle nachfolgenden Beispiele nur für INTERLIS 2.3 angegeben. Das Modellverzeichnis für INTERLIS 2.2 heisst ..\models22.

3.1 ICS for Windows

Die komfortabelste Bedienung bietet die Windowsoberfläche *ICS for Windows*. Sie können iG/Check über den Windows Start Knopf unter *Programme>infoGrips INTERLIS Tools>iGCheck for INTERLIS 2* aufrufen. Nach dem Start von *ICS for Windows* wird folgendes Menü angezeigt:



Die Funktionsknöpfe haben folgende Bedeutung:

- **`User Script...`** bzw. **`System Script...`** Konfigurationsdatei auswählen. Normalerweise wird automatisch *\user\script\igcheck\interlis2.cfg* geladen. Falls Sie eigene Konfigurationen erstellt haben, können Sie diese unter **`User Script...`** auswählen (s.a. Kapitel 5).
- **`Show Skript`**. Zeigt die aktuelle Konfigurationsdatei mit *notepad.exe* an.
- **`Run`**. iG/Check fragt zuerst den Namen der zu prüfenden Transferdatei ab, danach wird der Prüfvorgang gestartet. Ab iG/Check 2012 können auch mehrere Inputdateien gleichzeitig ausgewählt werden. Klicken Sie dazu mit gehaltener <CRTL> Taste auf die gewünschten Dateien.
- **`Show Log`**. Zeigt die erzeugte Logdatei mit *notepad.exe* an. Neben Meldungen und Fehlern wird am Schluss der Logdatei eine Statistik über alle gelesenen Objekte ausgegeben. Falls Sie mehrere Dateien ausgewählt haben, wird mit **`Show Log`** nur eine Zusammenfassung der Resultate angezeigt. Die detaillierten Ergebnisse werden in separaten .log Dateien mit dem gleichen Namen wie die Inputdateien abgelegt.

- `Cancel`. Verlässt *ICS for Windows*.

Die Anzeigefelder des *ICS for Windows* Menüs haben folgende Bedeutung:

- **iG/Script**. Zeigt das aktuell ausgewählte Skript an.
- **License Information**. Zeigt Lizenzinformationen an.
- **Status Information**. Unter **Inputobjects** wird die Anzahl der von iG/Check gelesenen Objekte angezeigt. Unter **Outputobjects** wird die Anzahl der in die XTF-Fehlerdatei geschriebenen Objekte angezeigt (s.a. 5.2). Unter **Errors** wird die Anzahl der gefundenen Fehler angegeben.

3.2 Bedienung im Batch-Modus

Im Batch-Modus kann iG/Check wie folgt aufgerufen werden:

```
igcheck2 <Transferdatei> [<Modelldatei1>]{+<ModelldateiN>}
```

Für `<Transferdatei>` geben Sie die zu prüfende INTERLIS-Transferdatei und für `<Modelldatei1..N>` die dazugehörigen INTERLIS-Modelldateien (*.ili*) an. Falls `<Modelldatei1..N>` weggelassen wird, sucht iG/Check die passenden Modelldateien automatisch in den Verzeichnissen `IGCHECK_DIR`\user\models23 und dann in `IGCHECK_DIR`\system\models23 (s.a. 5.1.2 `IGCHECK_PARAM.MODELS`).

Bemerkung: Für <Transferdatei> kann entweder ein relativer oder ein absoluter Dateipfad angegeben werden. Die <Modelldateien> werden jedoch immer relativ zu `MODEL_DIR` bzw. in `IGCHECK_DIR`\system\models23 oder in externen Internet Modell-Repositories gesucht (s.a. 5.1.4).

Meldungen und Fehler werden auf den Bildschirm bzw. in die Logdatei ausgegeben. Die Logdatei wird im gleichen Verzeichnis wie die Transferdatei angelegt und hat den gleichen Namen wie die Transferdatei mit der Ausnahme, dass die Endung der Datei von *.xtf* in *.log* abgeändert wird (z.B. Transferdatei = *test.xtf*, Logdatei = *test.log*). Neben Meldungen und Fehlern wird am Schluss der Logdatei eine Statistik über alle gelesenen Objekte ausgegeben. Bei mehreren Inputdateien wird eine Hauptlogdatei mit einer Zusammenfassung aller Resultate geschrieben. Die detaillierten Informationen findet man in einer separaten Logdatei pro Inputdatei.

Damit Sie iG/Check von der DOS Kommandozeile aus in einem beliebigen Verzeichnis starten können, müssen Sie die `PATH`-Variable um das `bin`-Verzeichnis von `ICS for Windows` erweitern. Die Variable können Sie wie folgt definieren:

- Öffnen Sie in der Windows-Systemsteuerung:

  ```
  System>Erweitert>Umgebungsvariablen
  ```

- Definieren oder erweitern Sie die Variable `PATH` als Systemvariable oder Benutzervariable um den Anteil

  ```
  c:\igcheck2\system\bin
  ```

  (Annahme: iG/Check wurde unter `c:\igcheck2` installiert).

- Melden Sie sich in Ihrem System ab und wieder an, oder starten Sie den PC neu.

4. Fehlermeldungen

4.1 Was überprüft wird

Wie bereits in der Einleitung beschrieben, überprüft iG/Check sowohl die INTERLIS-Definitionsdatei als auch die INTERLIS-Transferdatei. Die Modelldatei wird auf syntaktische Korrektheit analysiert. Dazu wird automatisch der offizielle INTERLIS Compiler der KOGIS aufgerufen (Version s.a. 1.1). Falls Fehler in der Modelldatei gefunden werden, wird der Prüfvorgang an dieser Stelle mit einer entsprechenden Fehlermeldung abgebrochen. Falls in der Modelldatei keine Fehler gefunden werden, wird als nächstes die Transferdatei analysiert. In der Transferdatei wird folgendes getestet:

- Syntaktische Korrektheit der Transferdatei, d.h. sind die XML-Tags `TRANSFER`, `HEADERSECTION`, `DATASECTION`, etc. in der richtigen Reihenfolge angeordnet, gibt es unbekannte XML-Tags etc.
- Korrektheit der Modell-, Ebenen- und Klassennamen.
- Statische Wertebereiche, d.h. alle Objektattribute haben Wertebereiche gemäss Modelldatei (inkl. Koordinatenbereiche von Punkten, Linien und Flächen).
- Dynamische Wertebereiche, d.h. Auflösung aller Referenzen (z.B. Referenz von Rolle `Gebaeudenummer.Objekt` auf Tabelle `BoFlaeche` in Topic `Bodenbedeckung`).
- Eindeutigkeit der Transferidentifikation und Basketidentifikationen pro Transferdatei.
- Kardinalität von BAG, LIST und Rollen.
- UNIQUE CONSTRAINT und LOCAL UNIQUE.
- Korrektheit der Topologie von AREA-Attributen (z.B. `BoFlaeche.Geometrie` im Topic `Bodenbedeckung`).
- Externe Referenzen (bei mehreren Inputdateien).
- EXISTENCE CONSTRAINT.
- OID AS UUIDOID.

4.2 Verhalten bei Fehlern

iG/Check prüft immer die ganze / alle Transferdatei(en) unabhängig wie viele oder welche Fehler gefunden werden. Bei syntaktischen Fehlern oder unbekannten Topic- bzw. Klassennamen synchronisiert sich iG/Check auf das nächste korrekte Element.

4.3 Fehlermeldungen

Fehlermeldungen werden von iG/Check in englischer Sprache ausgegeben. Die meisten Fehlermeldungen von iG/Check beziehen sich auf eine Zeile in der Transferdatei. Die erste Zeile der Transferdatei hat die Nummer 1. In der Transferdatei ist es möglich, dass sich ein Objekt über mehrere Zeilen erstreckt. Falls sich die Fehlermeldung auf ein mehrzeiliges Objekt bezieht, wird die Zeilennummer der ersten Zeile des fehlerhaften Objekts ausgegeben. Jede Fehlermeldung hat ausserdem eine Fehlernummer über

welche der Fehler eindeutig bestimmt werden kann. Im Anhang B ist die Bedeutung der Fehlermeldungen geordnet nach Fehlernummern zusammengestellt.

5. Konfigurationsparameter

Der Anwender kann die Ausgaben bzw. die Prüffunktionen von iG/Check über Konfigurationsparameter seinen Bedürfnissen anpassen. Die Konfigurationsparameter müssen in der Datei `IGCHECK_DIR`*\user\script\igcheck\interlis2.cfg* eingetragen werden. Diese Datei kann unter *ICS for Windows* mit der Funktion `Show Script` angezeigt und bearbeitet werden.

Hinweis: iG/Check for INTERLIS 2 unterstützt auch die ältere INTERLIS Version 2.2. Die meisten Parameter für die Version 2.2 bzw. Version 2.3 sind gleich und sind daher nur für die Version 2.3 beschrieben. Das Modellverzeichnis für INTERLIS 2.2 heisst ..\models22.

5.1 Allgemeine Parameter

5.1.1 IGCHECK_PARAM.FORCE_COMPILE

Falls dieser Parameter auf `ON` (Default = `OFF`) gesetzt wird, werden die INTERLIS Datenmodelle jedes Mal mit dem INTERLIS 2 Compiler analysiert. Falls die Option auf `OFF` gesetzt ist, wird zuerst nachgeschaut ob bereits eine vorcompilierte Version des Modells existiert (.ilo Datei). Das Laden von vorcompilierten Modelldateien ist viel schneller möglich, daher ist die Option normalerweise auf `OFF` gesetzt.

5.1.2 IGCHECK_PARAM.MODELS

Mit dieser Option kann man angeben, wie iG/Check die zum Datensatz gehörigen Modelldateien bestimmt. `XTF` bedeutet, dass die Modelle automatisch aus der gewählten *.xtf* Datei bestimmt werden. Dazu wird wie folgt vorgegangen:

- Es wird nach dem ersten `MODEL` Element der `HEADERSECTION` gesucht und aus diesem der Name des Hauptmodells gelesen. Falls kein `MODEL` Element gefunden wurde, wird nach dem letzten `ALIAS ENTRIES` Element der `HEADERSECTION` gesucht und aus diesem der Name des Hauptmodells gelesen. Falls keine `ALIAS ENTRIES` in der *.xtf* Datei vorhanden sind, wird der Modellname des Hauptmodells aus dem ersten Objekttag der `DATASECTION` extrahiert.
- Das Hauptmodell wird zuerst in `IGCHECK_DIR`\user\models23, dann in allfälligen externen Internet Modell-Repositories und schliesslich in `IGCHECK_DIR`\system\models23 gesucht (s.a. Anhang: Suche nach Modellen).
- Falls das Hauptmodell gefunden wird, wird das Hauptmodell nach `IMPORTS` durchsucht und so die allfälligen Basismodelle bestimmt. Für die Basismodelle wird ebenso verfahren.
- Der INTERLIS 2 Compiler wird automatisch mit den so bestimmten Modellen aufgerufen.

Normalerweise funktioniert das oben beschriebene Verfahren gut und ist ausserdem sehr praktisch in der Benutzung. Es kann aber sein, dass man in bestimmten Fällen die automatische Bestimmung der Datenmodelle ausschalten möchte. In diesem Fall kann man unter `MODELS` auch eine Liste von Datenmodelldateien in der korrekten Reihenfolge für den INTERLIS 2 Compiler angeben. Beispiel:

```
MODELS => units.ili,time.ili,coordsys.ili,dm01avch24d.ili
```

Die Dateien werden ebenfalls zuerst in IGCHECK_DIR\user\models23 dann in IGCHECK_DIR\system\models23 gesucht. Schliesslich hat man noch die Möglichkeit in der Datei IGCHECK_DIR*user\www\models\ilimodels.xml* spezielle Suchregeln zu definieren (s.a. Anhang).

5.1.3 IGCHECK_PARAM.MODEL_NAME

Unter MODEL_NAME kann man den Namen eines Basismodells des Hauptmodells angeben (z.B. DM01AVCH24D für eine kantonale Erweiterung). Die Überprüfung findet dann nur gemäss den Regeln des Basismodells statt (polymorpher Check). Hinweis: Dieses Feature ist in Version 1.1 noch nicht verfügbar.

5.1.4 IGCHECK_PARAM.MODEL_DIR

Normalerweise werden Modelldateien (.ili) zuerst in IGCHECK_DIR\user\models23 gesucht. Mit der Option MODEL_DIR kann man einen alternativen (User-)Suchpfad angeben (Default = OFF).

5.1.5 IGCHECK_PARAM.INPUT_EXTENSION

INTERLIS 2 Transferdateien weisen normalerweise die Endung *.xtf* auf. Falls man nun beim Öffnen der Inputdatei nach einer anderen Endung suchen möchte (z.B. *.xml*), kann man diese hier angeben (*ohne* Punkt).

5.1.6 IGCHECK_PARAM.VERSION

Mit VERSION kann die INTERLIS Version explizit angegeben werden. Mögliche Werte sind 2.2, 2.3 oder XTF (Default = XTF). Mit XTF wird die INTERLIS Version automatisch aus der Inputdatei bestimmt (VERSION Attribut der HEADERSECTION).

5.1.7 IGCHECK_PARAM.INTERNET_ACCESS

Mit INTERNET_ACCESS kann der Zugriff auf externe Internet-Repositories komplett abgeschaltet werden. Mögliche Werte sind ON oder OFF (Default = ON). Mit OFF werden Modelle nur in lokalen Verzeichnissen von iG/Check gesucht, wobei auch lokal gecachete Versionen von *.ili* Dateien aus den Internet-Repositories bei der Suche verwendet werden.

5.1.8 IGCHECK_PARAM.PROXY_*

Falls INTERNET_ACCESS auf ON gesetzt wurde, muss u.U. auch ein allfällig vorhandener HTTP Proxy-Server konfiguriert werden. Dafür stehen die Optionen PROXY_SERVER, PROXY_USER und PROXY_PASSWORD zur Verfügung. Falls Sie auf Internet-Repositories nicht zugreifen können, sollten Sie die obigen Parameterwerte bei Ihrem Netzwerkadministrator nachfragen.

5.2 Prüfungen ein- oder ausschalten

Die nachfolgenden Optionen können jeweils die Werte ON (= einschalten) oder OFF (= ausschalten) annehmen.

5.2.1 IGCHECK_PARAM.CHECK_AREA

Topologietest von Flächennetzen aktivieren (Default = ON).

5.2.2 IGCHECK_PARAM.CHECK_UNIQUE

UNIQUE CONSTRAINT prüfen (Default = ON).

5.2.3 IGCHECK_PARAM.MANDATORY_CONSTRAINT

MANDATORY CONSTRAINT prüfen (Default = ON). Hinweis: Benutzerdefinierte Funktionen können nicht automatisch getestet werden, da diese immer an spezielle Vereinbarungen (CONTRACTED) gebunden sind. Auf Anfrage ist die Implementierung von benutzerdefinierten Funktionen im Checker jedoch möglich.

5.2.4 IGCHECK_PARAM.CHECK_TAG_ORDER

Reihenfolge der XML-Tags prüfen (Default = ON). INTERLIS 2 schreibt eine vollständige Ordnung der XML-Tags in der Transferdatei vor. Es kann sein, dass eine Transferdatei sehr viele solche Fehler enthält. In diesem Fall kann man die Prüfung deaktivieren um die anderen Fehler besser erkennen zu können.

5.2.5 IGCHECK_PARAM.CHECK_EXTERNAL_REF

Prüft externe Referenzen (EXTERNAL) (Default = AUTO). Folgende Werte sind zulässig: ON, OFF und AUTO. Externe Referenzen sind Referenzen, welche auf Objekte in externen Dateien verweisen. Damit iG/Check diese Referenzen auflösen kann, müssen alle betroffenen Dateien gleichzeitig als Input ausgewählt werden. Im Modus AUTO werden externe Referenzen nur dann geprüft, wenn mehr als eine Inputdatei ausgewählt wurde.

5.3 Formatierung der Logdatei

5.3.1 DISPLAY_ATTRIBUTES

Standardmässig gibt iG/Check die Fehlernummer (s.a. Anhang B), die Fehlermeldung, die Zeilennummer und die Klasse des fehlerhaften Objekts aus. In gewissen Fällen möchte man jedoch mehr Informationen zum fehlerhaften Objekt direkt in der Fehlermeldung darstellen (z.B. die Nummer für LFP1). Mit der Map DISPLAY_ATTRIBUTES kann man pro Objektklasse die Attribute spezifizieren, welche in der Fehlermeldung ausgeben werden sollen, Bsp.:

```
MAP DISPLAY_ATTRIBUTES
   ! Topic,Class => Attributes
   FixpunkteKategorie1,LFP1 => LINE,CLASS,Nummer
   DEFAULT => LINE,CLASS
END_MAP
```

Im letzten Beispiel wird z.B. die Ausgabe der Attribute LINE, CLASS und Nummer für die Klasse FixpunkteKategorie.LFP1 verlangt. Für alle anderen Klassen (DEFAULT) wird nur LINE und CLASS ausgegeben (s.a. unten). Neben den Attributen gemäss Datenmodell stehen folgende vordefinierte Systemattribute zur Verfügung:

LINE: Zeilennummer des fehlerhaften Objekts.
CLASS: Klasse des fehlerhaften Objekts.
TOPIC: Topic des fehlerhaften Objekts.
MODEL: Modell des fehlerhaften Objekts.
TAG: XML-Tag des fehlerhaften Objekts.

5.3.2 IGCHECK_PARAM.MULTI_LINE_ERROR

Falls dieser Parameter auf ON (Default = OFF) gesetzt wird, wird jede Fehlermeldung in der Logdatei auf mehre Zeilen umgebrochen. Insbesondere zusammen mit der Map DISPLAY_ATTRIBUTES kann es sinnvoll sein diese Option zu aktivieren.

5.3.3 IGCHECK_PARAM.TRACE

Falls dieser Parameter auf ON (Default = OFF) gesetzt wird, wird jedes von iG/Check gelesene Objekt vollständig in der Logdatei aufgelistet.

5.3.4 IGCHECK_PARAM.TRACE_DRIVER

Falls dieser Parameter auf ON (Default = OFF) gesetzt wird, werden für jedes vom XTF Driver gelesene Objekt Meldungen in die Logdatei ausgegeben.

5.4 XTF Fehleroutput

5.4.1 IGCHECK_PARAM.XTF_ERRORLOG

Es besteht grundsätzlich die Möglichkeit, alle Fehlermeldungen auch als INTERLIS 2 XML Transferdatei auszugeben (.XTF). Falls die Option aktiviert ist (= ON), wird zusätzlich zur normalen Logdatei eine XTF-Fehlerdatei erzeugt. Die XTF-Fehlerdatei wird im gleichen Verzeichnis wie die Transferdatei angelegt und hat den gleichen Namen wie die Transferdatei mit der Ausnahme, dass die Endung der Datei von *.xtf* in *_err.xtf abgeändert wird* (z.B. Transferdatei = *test.xtf*, XTF-Fehlerdatei = *test_err.xtf*). Das zur Fehlerdatei gehörige INTERLIS 2 Datenmodell ist unter IGCHECK_DIR\system\models23\ErrorLog.ili abgelegt (s.a. Anhang C). Pro Fehler werden neben den Attributen Typ, Modul, Description auch die unter DISPLAY_ATTRIBUTES spezifizierten Attribute ausgegeben (in der Attributes Liste).

Anhang

A. Literaturverzeichnis

[1] Schweizer Norm SN 612 031, INTERLIS Modellierungssprache und Datentransfermethode, Ausgabe 2006.

[2] Eidg. Vermessungsdirektion, Datenmodell DM01AVCH24D, Ausgabe Juni 2004 (s.a. *http://www.cadastre.ch/pub/down/interlis/dm01avch24_de.pdf*).

B. Verzeichnis der Fehlermeldungen

1	**ili2c syntax error**
	Beschreibung: Der INTERLIS 2 Compiler konnte das zugehörige Datenmodell nicht compilieren. Weitere Informationen zu der Fehlermeldung sind aus der *.log* Datei ersichtlich.

2	**model parse error, <explanation>**
	Beschreibung: Die vom Compiler gelieferte Information konnte von iG/Check nicht vollständig verarbeitet werden. Der Prüfvorgang wurde abgebrochen. Weitere Informationen dazu stehen in der *.log* Datei.
	Beispiel: `model DM01AVCH24D contains errors, see also ..\user\models23\dm01avch24d.err`

3	**xtf error, <explanation>**
	Beschreibung: Fehlermeldung, welche durch den XTF Driver gemeldet wird. Die Art der Fehlermeldung wird in <explanation> näher erläutert.
	Beispiel: `xtf error, Attribute OPERATION has wrong value in object`

4	**fatal syntax error in input file**
	Beschreibung: Der XTF Driver kann die Inputdatei nicht verarbeiten. Der Prüfvorgang wird abgebrochen. Weitere Information dazu sind aus der *.log* Datei ersichtlich.

10	**unable to convert <attr> to type <basetype>**
	Beschreibung: Der Wert des Attributs <`attr`> kann nicht in den Basistyp <`basetype`> übersetzt werden. Bemerkung: für die weiteren Tests des Attributs wird für den Wert des Attributs `UNDEFINED` angenommen.

	Beispiele:
	`unable to convert Geometrie to type COORD`
	`unable to convert LFP1Symbol_von to type LINK`

11	**<attr|role> has to be defined**
	Beschreibung: Der Wert des Attributs oder der Rolle <`attr|role`> ist undefiniert. Das Attribut oder die Rolle ist aber im Datenmodell als `MANDATORY` (obligatorisch) definiert.

	Beispiele:
	`Geometrie has to be defined`
	`Nummer has to be defined`

12	**the value of atttribute <attr> is out of range, <explanation>**
	Beschreibung: Der Wert des Attributs <attr> ist ausserhalb des im Modell spezifizierten Wertebereichs. Dieser Fehlertyp kennt einige Untervarianten. Z.B. gehört die Verletzung des Koordinatenbereichs durch einen Punkt ebenso dazu wie die Benutzung von unerlaubten Zeichen in Textattributen. Unter <explanation> werden weitere Hinweise zur konkreten Art der Fehlermeldung gegeben.

	Beispiele:
	`the value of NBIdent is out of range, text is too long 18 > 12`
	`the value of LageGen is out of range, 900.0 > 700.0`

13	**invalid linesegment <coord1> <coord2>**
	Beschreibung: Das Liniensegment ist keine gültige Liniendefinition. Beide Punkte sind identisch.
	Beispiele: `invalid linesegment 1.0/1.0/0.0 1.0/1.0/0.0`

14	**invalid arcsegment <coord1> <coord2> <coord3>**
	Beschreibung: Der Kreisbogen ist keine gültige Kreisbogendefinition. Zwei oder drei Punkte sind identisch oder der mittlere Punkt liegt auf der Geraden durch den 1. bzw. 3. Punkt.
	Beispiele: `invalid arcsegment 0.0/0.0 1.0/1.0 2.0/2.0`

15	**the value of <role/refattr> is out of range, <explanation>**
	Beschreibung: Die Referenz <role/refattr> zeigt auf eine TID, welche in der Transferdatei nicht vorkommt.
	Beispiele: `the value of Entstehung is out of range, no object found with tid=zh04p2t400000020`

16	**unknown attribute/link <attr> in <class/structure>**
	Beschreibung: Das Attribut/der Link <attr> darf in der Klasse/Struktur <class/structure> nicht vorkommen.
	Beispiele: `unknown attribute Geometr in DM01AVCH24D.FixpunkteKategorie1.LFP1`

17	**attributes are out of order for <class/structure>, <explanation>**
	Beschreibung: Die Attribute der Klasse/Struktur <class/structure> sind nicht in der richtigen Reihenfolge in der Transferdatei angeordnet. In <explanation> ist die korrekte Reihenfolge angegeben, dabei werden optionale Attribute zwischen [] dargestellt.
	Beispiele: `attributes are out of order for DM01AVCH24D.FixpunkteKategorie2.LFP2, correct order is Geometrie,LageGen,LageZuv,NBIdent,Nummer, [HoeheGeom],[HoeheGen],[HoeheZuv],[Punktzeichen], [Begehbarkeit],Entstehung`

18	**unknown topic <topic>, object in line <line> ignored**
	Beschreibung: Das Topic <`topic`> des Objekts ist unbekannt. Das Objekt kann nicht weiter überprüft werden.
	Beispiele: `unknown topic FixpunkteKtegorie1, object in line 823 ignored`

19	**<struct> is no extension of type <basestruct>, structure ignored**
	Beschreibung: Die Struktur <`struct`> ist keine zulässige Erweiterung von <`basestruct`>. <`struct`> wird nicht weiter geprüft.
	Beispiele: `QPA_de.QPA.Gewaesserabschnitt is no extension of type QPA_de.QPA.Grenze, structure ignored`

20	**duplicate transferid, <explanation>**
	Beschreibung: Die Transferid (`TID` oder `BID`) kommt in der Transferdatei mehrfach vor.
	Beispiele: `duplicate transferid, TID=ch037kym00490000000200110022`

21	**the value of <role/attr> is out of range, link <key> points to invalid target class <class>**
	Beschreibung: Die Referenz `<role/attr>` zeigt auf ein Objekt mit der falschen Klasse <`class`>.
	Beispiele:

22	**list/bag <list/bag> has invalid cardinality <count>, <explanation>**
	Beschreibung: Die Anzahl Elemente <`count`> des `<list/bag>` Attributs ist falsch.
	Beispiele: `list/bag Grenze has invalid cardinality 1, valid range is 2..2`

23	**unknown class <class>, object in line <line> ignored**
	Beschreibung: Die Klasse <`class`> des Objekts ist unbekannt. Das Objekt kann nicht weiter überprüft werden.
	Beispiele: `unknown class LfP1, object in line 324 ignored`

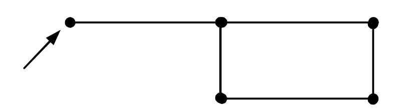
24	**invalid tag <tag>, object in line <line> ignored**
	Beschreibung: Das Tag <`tag`> des Objekts ist unbekannt. Das Objekt kann nicht weiter überprüft werden.
	Beispiele: `unknown tag abc, object in line 323 ignored`

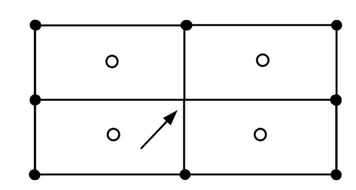
50	**cardinality <card> of role <role>[<ref>] is invalid, <explanation>**
	Beschreibung: Die Kardinalität <card> der Rolle <role> ist ausserhalb des gültigen Bereichs. Im Beispiel unten zeigen z.B. zwei LFP1Symbol Objekte auf das LFP1 Objekt mit der TID ch04p2t400000021. Erlaubt ist aber max. 1 Verweis.
	Beispiele: cardinality 2 of role LFP1Symbol.LFP1Symbol_von[ch04p2t400000021] is invalid, valid range is 0..1

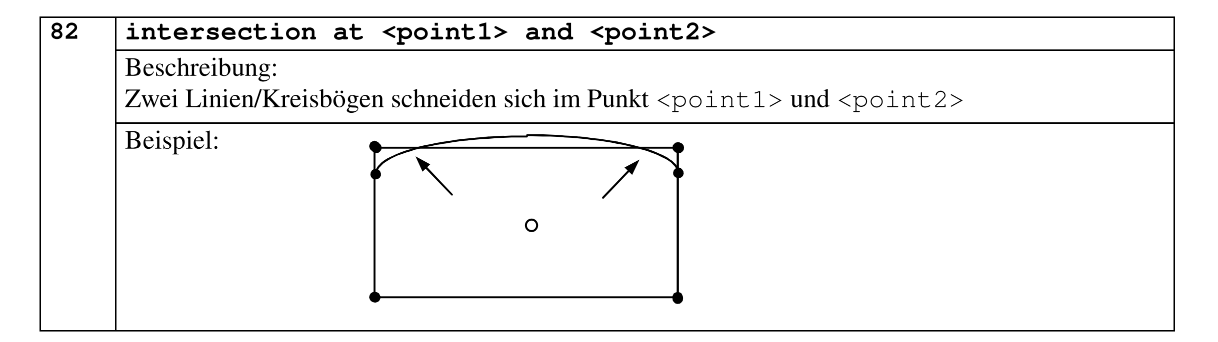
60	**MANDATORY constraint <constraint> failed for object <class>**
	Beschreibung: Der MANDATORY constraint <constraint> ergab den Wert false. Nähere Hinweise zu <constraint> findet man unter IGCHECK_DIR\\<*model*>*.ilp*.
	Beispiele: MANDATORY constraint CONSTRAINT_1 failed for object test_ce1.Topic.ClassB

70	**UNIQUE constraint <attributes> (values=<values>) violated, <hint>**
	Beschreibung: Die Attributkombination <attributes> kommt mehr als einmal in der Transferdatei vor.
	Beispiele: UNIQUE constraint NBIdent,Identifikator (values=0,TS 1) violated, see also line 28139

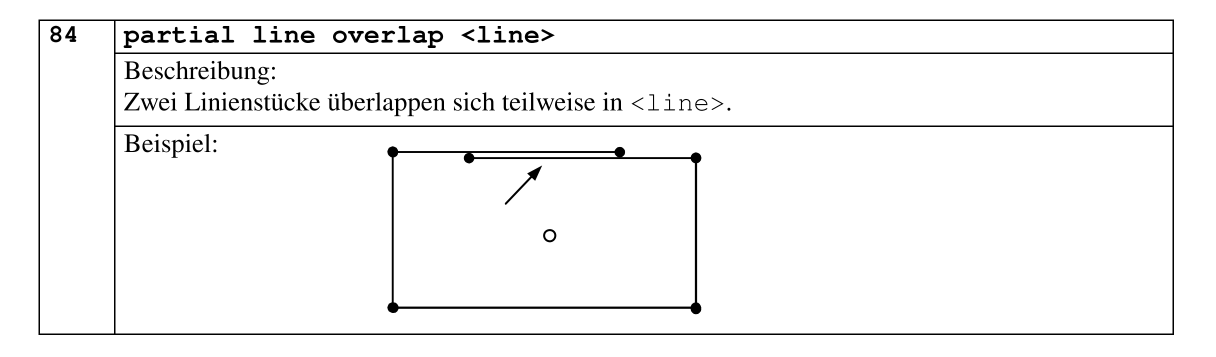
71	**LOCAL UNIQUE constraint <path>:<key> violated, <hint>**
	Beschreibung: Die Attributkombination <key> kommt mehr als einmal in der Liste oder dem Bag <path> vor.
	Beispiele: LOCAL UNIQUE attr21:attr00 violated, see also attr00=a

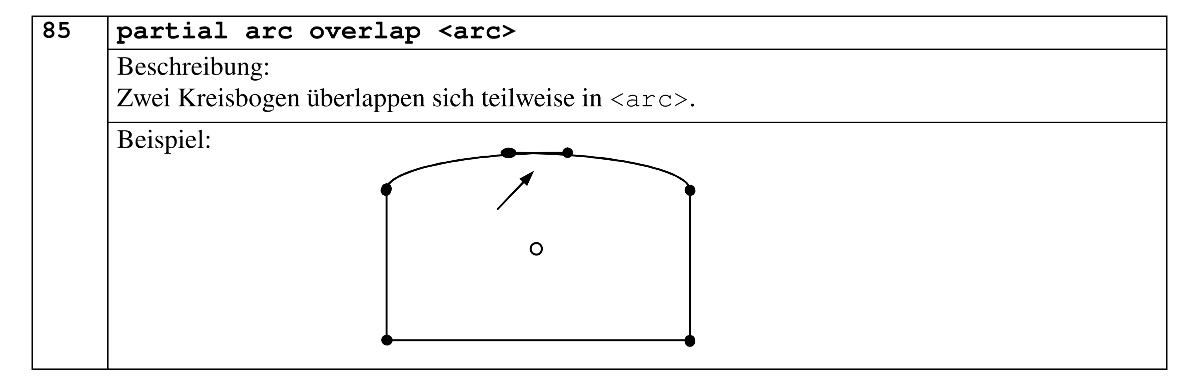
80	**open node at <point>**
	Beschreibung: An der angegebenen Stelle <point> befindet sich ein sog. 1er-Knoten. 1er-Knoten sind Punkte, die Start- oder Endpunkt von nur einer Linie sind.
	Beispiel: 

81	**intersection at <point>**
	Beschreibung: Zwei Linien/Kreisbögen schneiden sich im Punkt <point>
	Beispiel: 

82	**intersection at <point1> and <point2>**
	Beschreibung: Zwei Linien/Kreisbögen schneiden sich im Punkt `<point1>` und `<point2>`
	Beispiel: 

83	**invalid connecting edge <line>**
	Beschreibung: Das angegebene Linienstück `<line>` gehört zu einer ungültigen Verbindungslinie (Nabelschnur). Eine Nabelschnur ist eine Verbindungslinie zwischen dem äusseren Perimeter und einer Insel der Fläche.
	Beispiel:

84	**partial line overlap <line>**
	Beschreibung: Zwei Linienstücke überlappen sich teilweise in `<line>`.
	Beispiel: 

85	**partial arc overlap <arc>**
	Beschreibung: Zwei Kreisbogen überlappen sich teilweise in `<arc>`.
	Beispiel: 

86	**full line overlap <line>**
	Beschreibung: Das Linienstück `<line>` liegt vollständig auf einem anderen Linienstück.
	Beispiel:

87	**full arc overlap <arc>**
	Beschreibung: Der Kreisbogen `<arc>` liegt vollständig auf einem anderen Kreisbogenstück.
	Beispiel:

88	**invalid node at <point>**
	Beschreibung: Es wurde ein "unechter" Knoten in einem Flächennetz gefunden. Eine Linie 1 stösst an eine Linie 2 ohne dass die Linie 2 an der Stelle einen Knoten besitzt.
	Beispiel:

999	**internal error, <explanation>**
	Beschreibung: Es ist ein interner Fehler in iG/Check aufgetreten. Der Grund wird in <explanation> erläutert. Nach einem internen Fehler wird der Prüfvorgang abgebrochen, d.h. es werden keine weiteren Tests mehr durchgeführt. Bitte schicken Sie die Transferdatei und die Modelldateien an info@infogrips.ch.
	Beispiele: `internal error, enumeration type has no VALUES component`

C. Datenmodell der Fehlerdatei (*_err.xtf*)

```
!!----------------------------------------------------------------
!! Error Output iG/Check for INTERLIS 2
!!----------------------------------------------------------------

INTERLIS 2.3;

MODEL ErrorLog (en) AT "http://www.infogrips.ch/models"
   VERSION "2007-10-03" =

   IMPORTS UNQUALIFIED INTERLIS;

   DOMAIN
      Coord2D = COORD
         480000.000 .. 850000.000 [m],
         70000.000 .. 310000.000 [m],
         ROTATION 2 -> 1;

  TOPIC Errors =

     DOMAIN

        STRUCTURE AbstractGeometry (ABSTRACT) =
        END AbstractGeometry;

        STRUCTURE PointGeometry EXTENDS AbstractGeometry =
           Geom: MANDATORY Coord2D;
        END PointGeometry;

        STRUCTURE LineGeometry EXTENDS AbstractGeometry =
           Geom: MANDATORY POLYLINE WITH (STRAIGHTS,ARCS)
                           VERTEX Coord2D;
        END LineGeometry;

        STRUCTURE SurfaceGeometry EXTENDS AbstractGeometry =
           Geom: MANDATORY SURFACE WITH (STRAIGHTS,ARCS)
                           VERTEX Coord2D;
        END SurfaceGeometry;

        STRUCTURE Attribute =
           Name: TEXT*40;
           Value: TEXT*255;
        END Attribute;

     CLASS Error =
        Number: MANDATORY 1 .. 1000000;
        Module: MANDATORY TEXT*20;
        Description: MANDATORY TEXT*255;
        Model: TEXT*80;
        Topic: TEXT*80;
        Class: TEXT*80;
        Tid: TEXT*40;
        Line: 1 .. 1000000000;
        Geom: LIST {0..*} OF AbstractGeometry;
        UserAttributes: LIST {0..*} OF Attribute;
     END Error;

  END Errors;

END ErrorLog.
```

D. Suche nach Modelldateien

Mit der iG/Check Version 2012 wurde die Möglichkeit der Suche nach Modellen nochmals stark erweitert. Hier alle wichtigen Punkte im Überblick:

- Weiterhin wird die automatische Suche nach Modellen in den Verzeichnissen IGCHECK_DIR\user\models.. bzw. IGCHECK_DIR\system\models.. unterstützt. Damit ein Modell <Modell> auch gefunden wird, muss die zughörige Modelldatei obligatorisch <Modell>*.ili* heissen.
- Mit dem Parameter IGCHECK_PARAM.MODELS kann auch explizit eine Liste von Modelldateien angegeben werden.
- Zusätzlich können Modelldateien, welche nicht gleich heissen wie das Modell, in der XML-Datei IGCHECK_DIR*\user\www\models\ilimodels.xml* eingetragen werden. Die XML-Datei muss gemäss dem INTERLIS 2.3 Datenmodell IliRepository09 aufgebaut sein. Nachfolgend ein kleines Beispiel dazu:

```
<IliRepository09.RepositoryIndex BID="b1">
   <IliRepository09.RepositoryIndex.ModelMetadata TID="1">
      <Name>CoordSys</Name>
      <SchemaLanguage>ili2_3</SchemaLanguage>
      <File>models23/CoordSys-20050616.ili</File>
      <Version>2003-03-18</Version>
   </IliRepository09.RepositoryIndex.ModelMetadata>
</IliRepository09.RepositoryIndex>
```

  Hier wird z.B. die INTERLIS 2.3 Modelldatei *models23\CoordSys-20050616.ili* für das Datenmodell CoordSys eingetragen. Achtung: Das obige Beispiel ist nicht vollständig. Es fehlt z.B. die HEADER bzw. die DATA Section.

- Schliesslich ist es möglich iG/Check automatisch in externen Internet Modell-Repositories suchen zu lassen. Dazu muss die Datei IGCHECK_DIR*\user\www\models\ilisite.xml* gemäss INTERLIS 2.3 Datenmodell IliSite09 angepasst werden. Nachfolgend ein Ausschnitt aus der Standard *ilisite.xml* Datei von iG/Check:

```
<subsidiarySite>
   <IliSite09.RepositoryLocation_>
      <value>user</value>
   </IliSite09.RepositoryLocation_>
   <IliSite09.RepositoryLocation_>
      <value>http://models.interlis.ch</value>
   </IliSite09.RepositoryLocation_>
   <IliSite09.RepositoryLocation_>
      <value>system</value>
   </IliSite09.RepositoryLocation_>
</subsidiarySite>
```

  Die obige Definition bedeutet, dass iG/Check zuerst im lokalen *user* Verzeichnis, dann im Internet Repository *http://models.interlis.ch* und schliesslich im lokalen *system* Verzeichnis nach Modelldateien sucht. Im Prinzip kann die obige Liste beliebig ergänzt oder verändert werden. Das sollte jedoch in den wenigsten Fällen notwendig sein, da auch via *http://models.interlis.ch* verknüpfte Modell-Repositories automatisch durchsucht werden.

- Falls mit Internet Modell-Repositories gearbeitet wird, werden die Modelldateien nur dann frisch herunter geladen, wenn die Modelldateien geändert haben. Die geladenen Dateien werden unter *\user\models..\cache* lokal gespeichert. Ob eine

Änderung stattgefunden hat oder nicht, wird via den MD5-Hash der *.ili* Datei entschieden.